北京航空航天大學
Beihang University

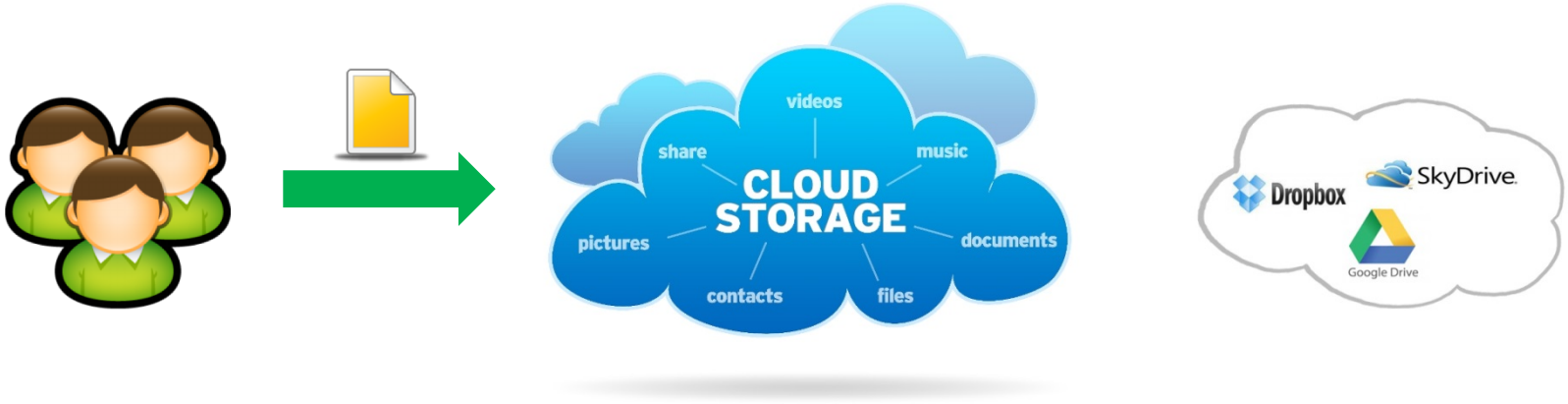
Extending the CP-ABE Scheme for Supporting Flexible Access Control

Bo Lang

24 July, 2013

Background

- The importance of data self-protection capability



Integrity & Confidentiality

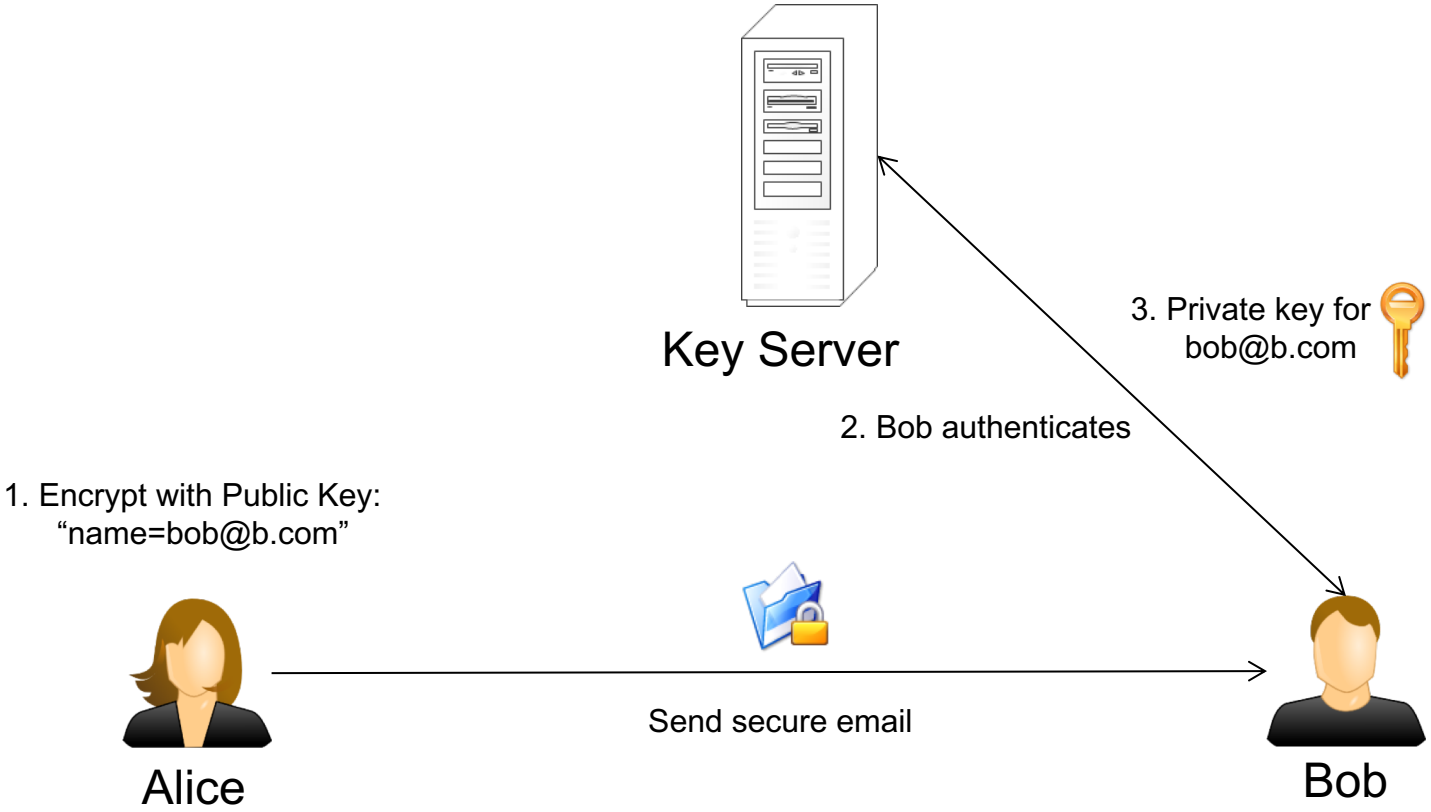
Reliable ?

Access Control & encryption

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

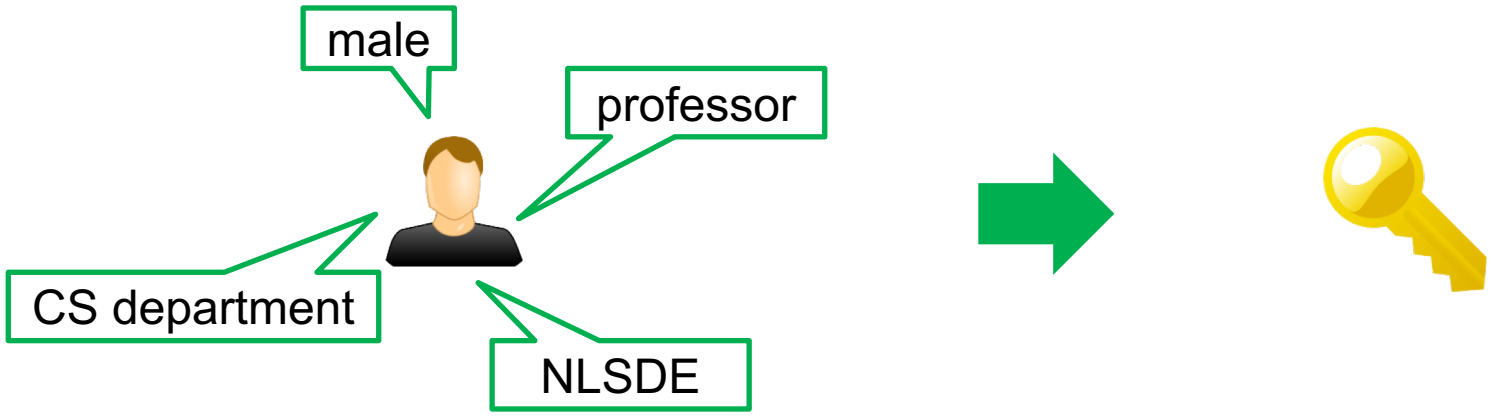
Background

- Identity Based Encryption



Background

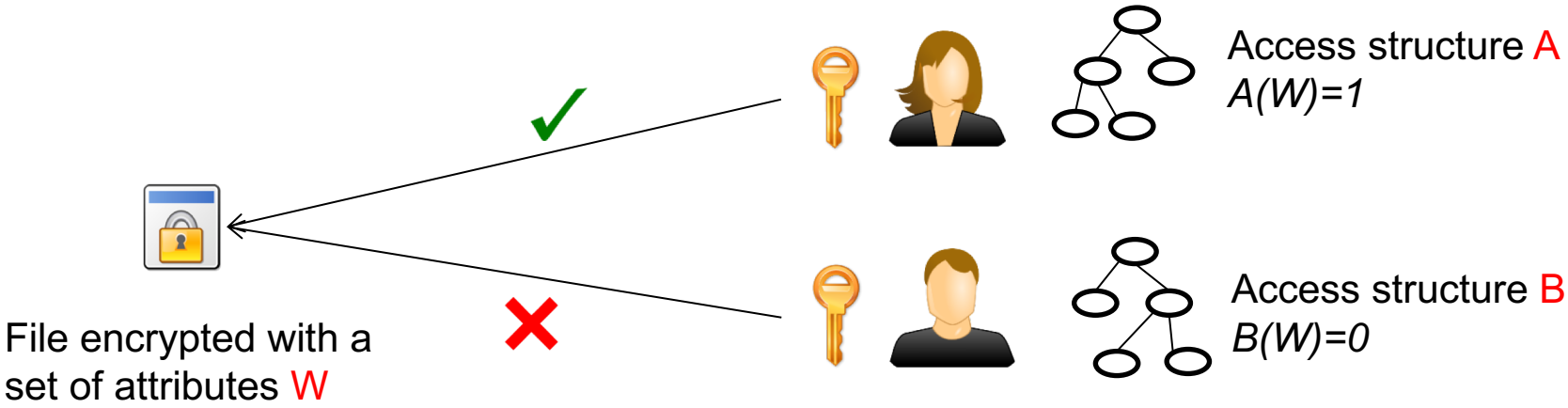
- Attribute Based Encryption
  - ✓ An extend scheme of Identity based Encryption
  - ✓ Utilization of attribute information for Encryption/Decryption
  - ✓ Dynamically control the user group of the encrypted data



State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Background

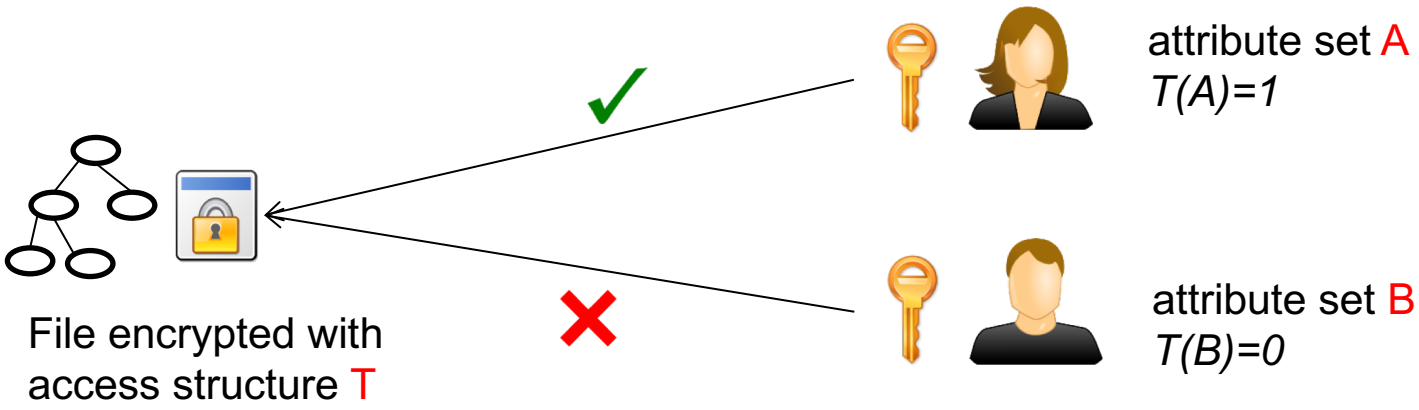
- Key Policy Attribute based Encryption (Goyal et al., 2006)



State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

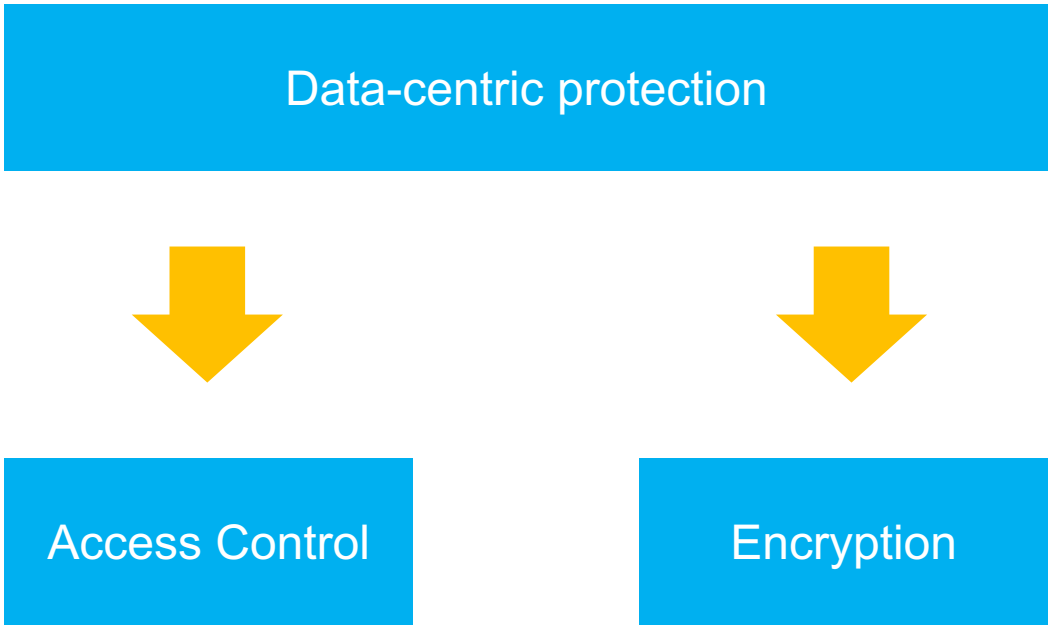
Background

- Ciphertext Policy Attribute based Encryption (Bethencourt et al., 2007)



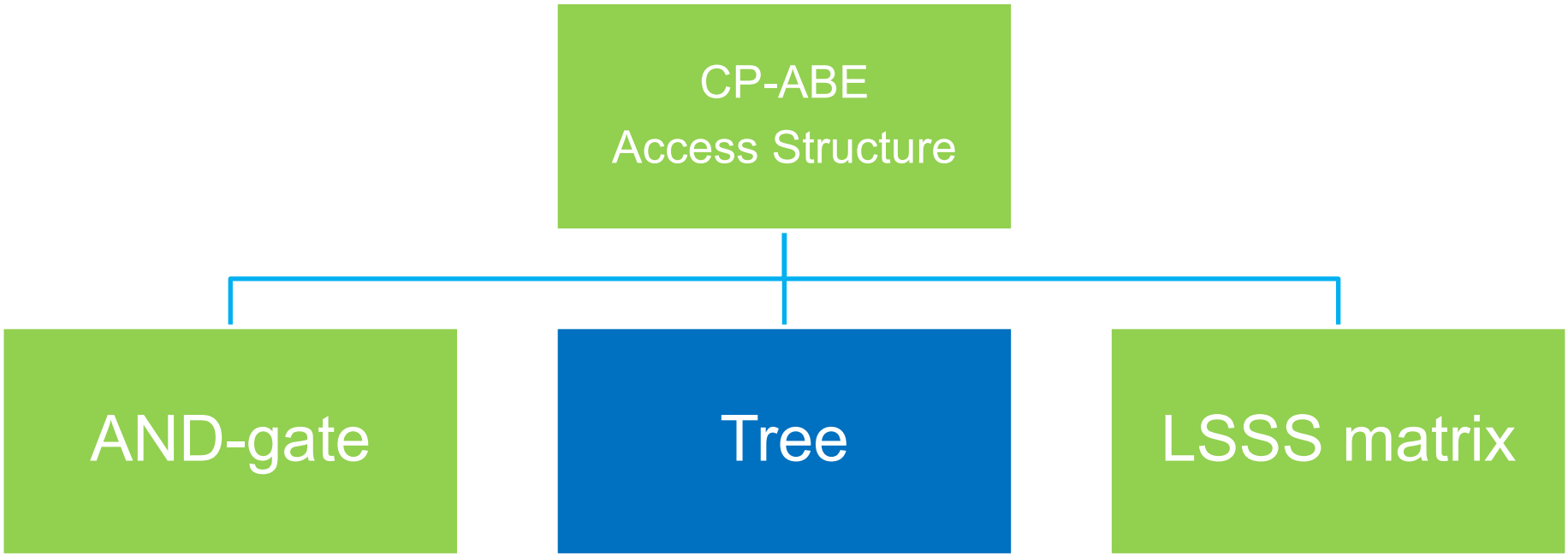
State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Background



State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Background



State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Related Work

- CP-ABE Schemes
  - ✓ BSW07
    - The initial structure of CP-ABE
    - Security:
      - General group model rather than the standard numerical theoretical assumptions
    - Expressive ability : Tree Structure
      - AND, OR and threshold operations
      - “bag of bits” for express policies containing <, ≤, >, ≥
      - Do not support “NOT”operator
  - ✓ CN07
    - Security
      - CPA security under DBDH assumption
    - Expressive ability :
      - AND, NOT opeartions
  - ✓ BCP-ABE
    - Security
      - CPA security under DBDH assumption
    - Expressive ability : Tree Structure
      - AND, OR and threshold operations

Related Work

- CP-ABE Schemes
  - ✓ BCP-ABE
    - Security
      - CPA security under DBDH assumption
    - Expressive ability : Tree Structure
      - AND, OR and threshold operations
    - Improvement of Liang et al
      - Shorten the system's public key
      - Shorten the user's private key
      - Shorten the length of the ciphertext
  - ✓ NYO08
    - partially hidden access structures
    - Expressive ability : And-gate
      - Supported the AND operation
      - Attributes have more than one candidate value
    - Improvement of Emura et al
      - Constant ciphertext length

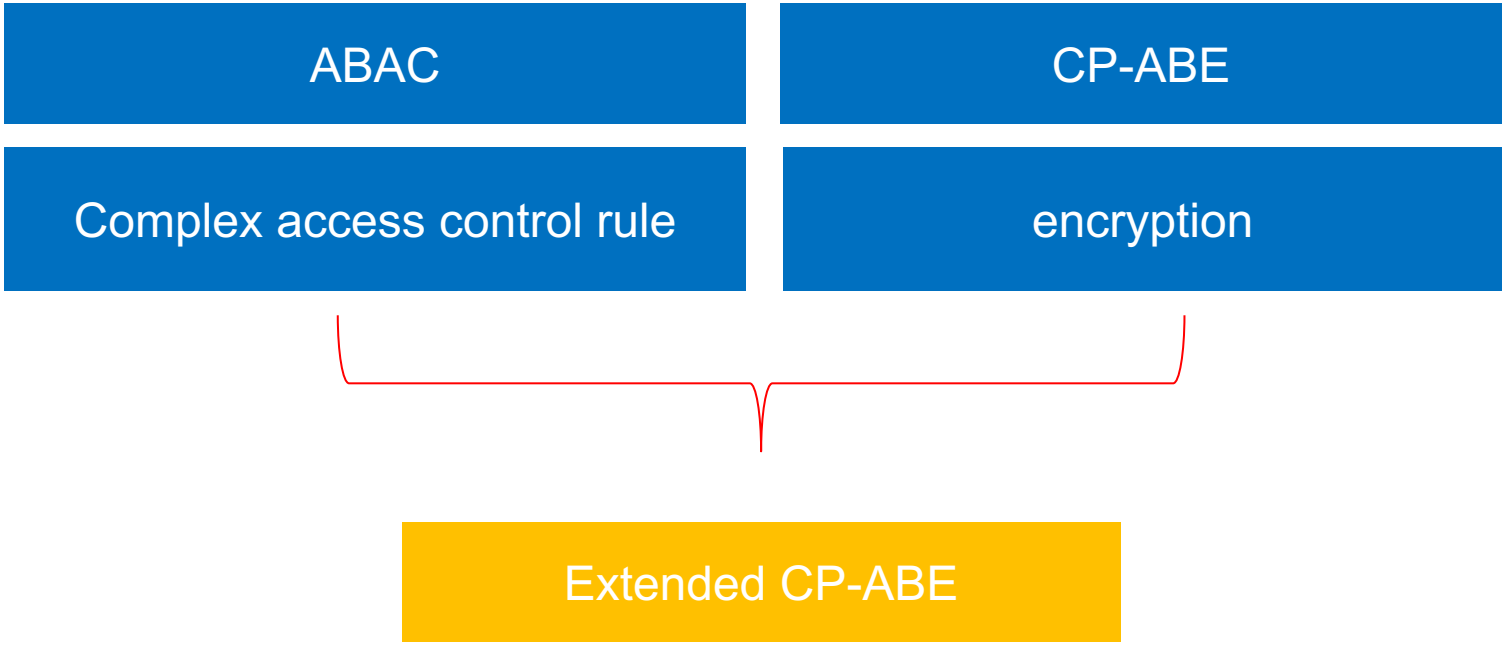
Related Work

- CP-ABE Schemes
  - ✓ ITHJ09
    - Security
      - CPA security under DBDH assumption
    - Expressive ability : Tree Structure
      - AND, OR and threshold operations
    - the costs of key generation, encryption and decryption are lower than BSW07
  - ✓ Waters08
    - Expressive ability : LSSS matrix
    - Improvement of Lewko et al
      - supported any monotone access formula

Related Work

- Others Schemes
  - ✓ Junod and Karlov
    - Support complex Boolean access policies
  - ✓ Chen et al
    - Support a non-monotone AND gate policy
  - ✓ Attrapadung et al
    - Support non-monotonic access structures and with constant ciphertext size
  - ✓ Zhiguo et al
    - Support hierarchical attribute-set-based encryption
- Conclusion
  - ✓ Only W08 and ITHJ09 support the AND, OR and threshold operation
    - Under the theoretical assumptions of the standard numerical
    - The computation cost of ITHJ09 is lower than W08

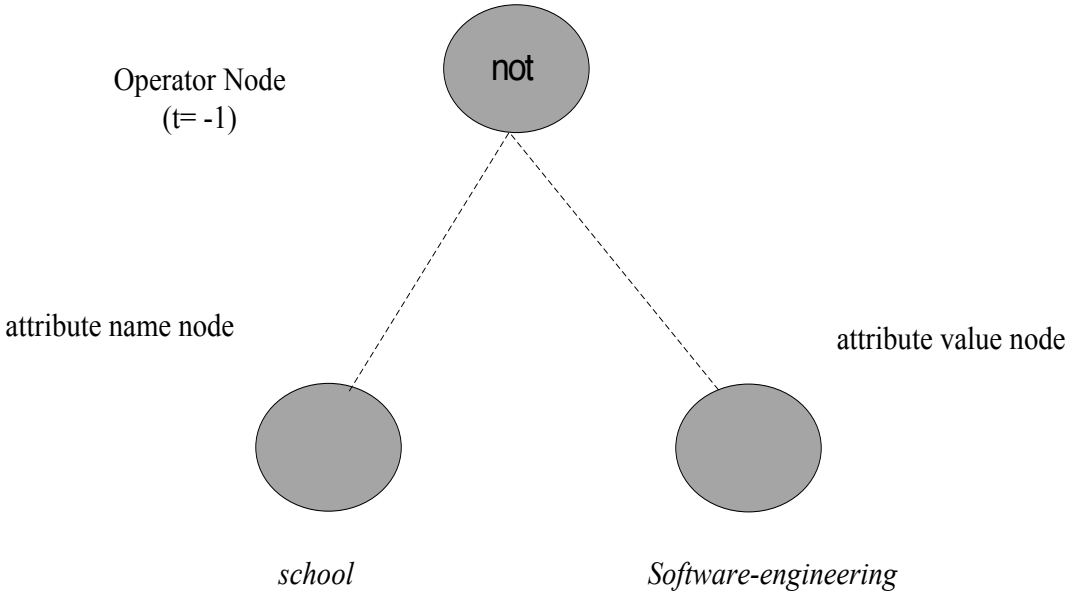
Our Contribution



State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

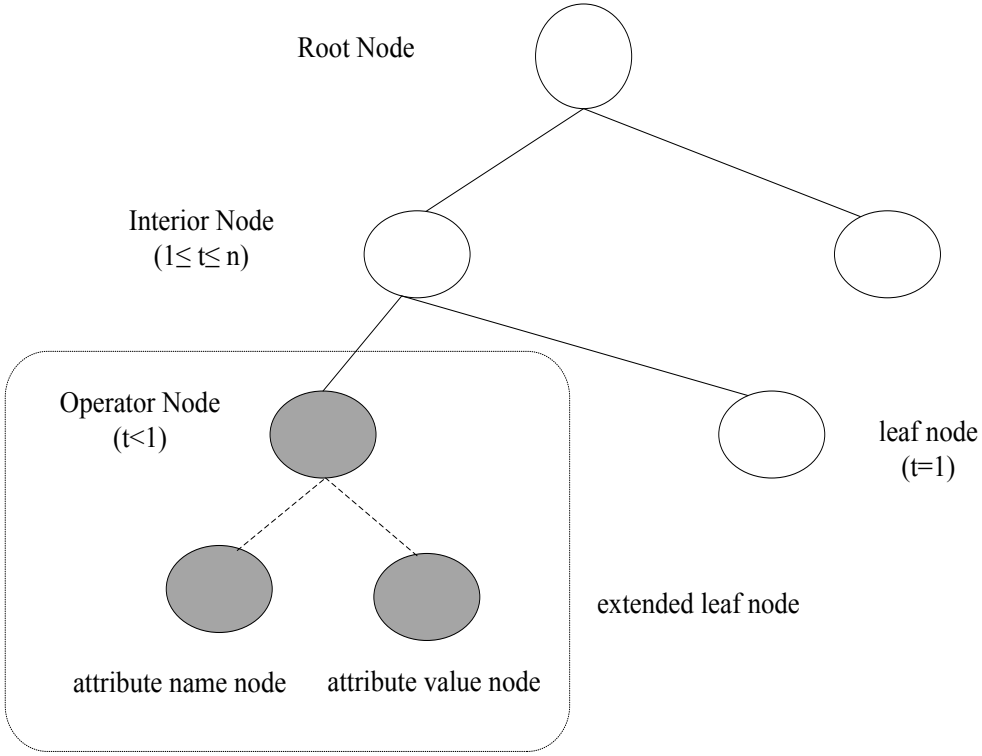
Extended Leaf Node

- The example of extend leaf node



Extended Leaf Node

- The example of extend access tree



Extended Leaf Node

- The values of t

Table 1: Values of *t* and its corresponding operator.

Value t	Operator
-1	not
-2	<
-3	>
-4	≤
-5	≥

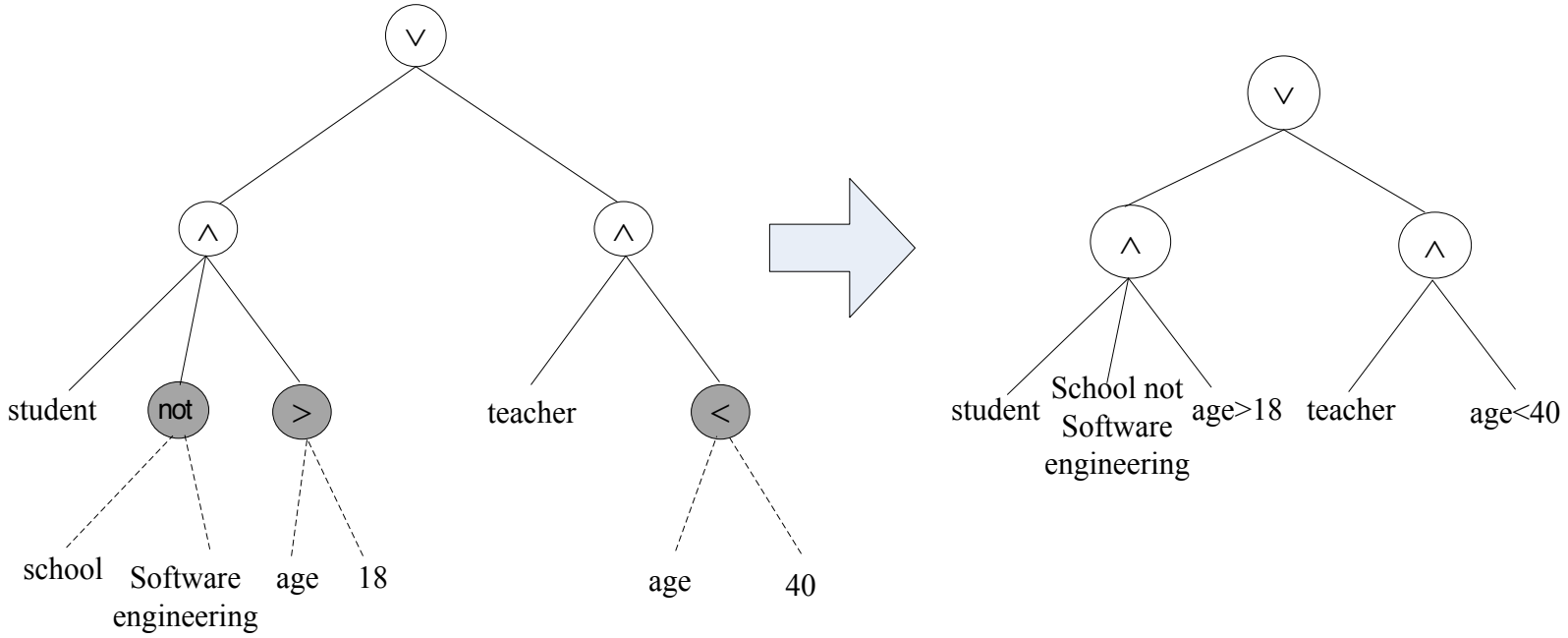
Transform the extended policy tree

ALGORITHM 1: Attribute Verification	
1:	Get the expression exp(N.O.V) of the extended leaf node, where N, O and V denote the basic attribute name, the operator and the attribute value respectively;
2:	Traverse the basic attribute set A' to find the basic attribute N and its value V';
3:	Let N=V', and calculate the expression exp(N.O.V);
4:	if the value of exp(N.O.V) is true
5:	Convert exp(N.O.V) to string S="N.O.V"
6:	return S;
7:	else
8:	return null;
9:	end if

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Transform the extended policy tree

- Example of transforming the extended policy tree



ECP-ABE specific scheme

- **Initialize**
  - ✓ Generate public parameter *pk* and master key *mk*
    - Generate a bilinear group G of prime order p with a generator g and a bilinear map e :$G \times G \rightarrow G_T$.
    - Generate the attribute set U = $\{a_1, a_2, \ldots, a_m\}$, for some integer m, and random elements $\alpha, t_1, t_2, \ldots, t_m \in Z_p^*$.
    - Let $y = e(g,g)^{\alpha}$, $T_j = g^{tj}$ $(1 \le j \le m)$.
    - The public key is $pk = (e, g, y, T_j (1 \le j \le m))$, and the master key is $mk = (\alpha, t_j (1 \le j \le m))$.
  - ✓ Give *pk* to the encryption party.
- **Specify the access policy**
  - ✓ the encryption party specifies access policy, which is expressed by an extended tree T^*.

ECP-ABE specific scheme

Encryption

- ✓ Convert the T^* to the standard tree T
- ✓ Select a random element $s \in Z_p^*$ and compute $c_0 = g^s$ and $c_1 = M \cdot y^s = M \cdot e(g,g)^{\alpha s}$
- ✓ Set the value of the root node of T to be s
- ✓ Mark all child nodes as un-assigned, and mark the root node assigned
- ✓ Recursively, for each un-assigned non-leaf node, do the following:
  - If its child nodes are un-assigned, the secret s is divided using (t,n)-Shamir secret sharing technique. The relation of n and t is:
    - if the symbol is of then $1 < t < n$;
    - if the symbol is AND, then $t = n$;
    - if the symbol is OR, then $t = 1$.
  - To each child node a share secret $s_i = f(i)$ is assigned
  - Mark this node assigned.
- ✓ For each leaf attribute $a_{j,i} \in T$, compute $c_{j,i} = T_j^{s_i}$.
- ✓ Return the ciphertext: $C_T = (T, c_0, c_1, \forall a_{j,I} \in T : c_{j,i})$.

ECP-ABE specific scheme

- **Secret key request**
  - ✓ Analyze the structure of T^* and find the extended parts
  - ✓ Give PKG his basic attribute set w and the extended parts of the access tree
- **Secret key generation**
  - ✓ Verify the basic attribute
  - ✓ Extract the attribute name, the attribute value and the operator
  - ✓ Run *Attribute Verification Algorithm* to get the new attribute set w^*
  - ✓ Generate the secret key sk_{w^*} corresponds to w^*
    - Select a random value $r \in Z_p^*$, $d_0 = g^{\alpha - r}$.
    - For each attribute a_j in w, compute $d_j = g^{rt_j^{-1}}$
  - ✓ Send key back to the user

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

ECP-ABE specific scheme

- **Decryption**
  - ✓ Select the smallest attribute set $w' \in w^*$ that corresponds to sk_{w^*} satisfies T
  - ✓ For every attribute $a_j \in w'$, computing:

$$m = \frac{c_1}{e(c_0, d_0) \cdot \prod_{a_j \in w'} e(c_{j,i}, d_j)^{l_i(0)}}$$

  - ✓ $l_i(0)$ is a Lagrange coefficient.

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

ECP-ABE specific scheme

- **Correctness Proof**:

$$m' = \frac{c_1}{e(c_0, d_0) \cdot \prod_{a_j \in w'} e(c_{j,i}, d_j)^{l_i(0)}}$$

$$= \frac{m \cdot e(g,g)^{\alpha s}}{e(g^s, g^{\alpha - r}) \cdot e(T_j^{s_i}, g^{rt_j^{-1}})^{l_i(0)}}$$

$$= \frac{m \cdot e(g,g)^{\alpha s}}{e(g^s, g^{\alpha - r}) \cdot \prod_{a_j \in w'} e(g^{t_j s_i}, g^{rt_j^{-1}})^{l_i(0)}}$$

$$= \frac{m \cdot e(g,g)^{\alpha s}}{e(g^s, g^{\alpha - r}) \cdot e(g,g)^{rs}}$$

$$= \frac{m \cdot e(g,g)^{\alpha s}}{e(g^s, g^{\alpha})}$$

$$= m$$

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

ECP-ABE Performance analysis

- IND-sAtt-CPA game
  - ✓ **Init.**
    - The adversary A chooses the challenge access tree T^* and gives it to the challenger, T^* is an extended tree.
  - ✓ **Setup.**
    - The challenger runs Setup to generate(pk,mk) and gives the public key pk to adversary A.
    - The challenger also transforms T^* to the equivalent standard tree T.
  - ✓ **Phase1.**
    - Adversary A makes a secret key request to the Keygen oracle for any attribute set $w = \{ a_j \mid a_j \in U\}$, with the restriction $a_j \notin T^*$ and a_j does not satisfy the policy attribute requirement expressed by the extend part of T^*.
    - The challenger runs *Attribute Verification Algorithm* to generate extended attribute set w^* and then returns Keygen(w^*, mk).

ECP-ABE Performance analysis

- IND-sAtt-CPA game
  - ✓ **Challenge.**
    - Adversary A sends to the challenger two equal length messages m_0, m_1.
    - The challenger picks a random bit $b \in \{0,1\}$ and returns $C_b = \text{Encrypt}(m_b, T^*, pk)$.
  - ✓ **Phase2.**
    - Adversary A can continue querying Key-generation oracle with the same restriction as in **Phase1**.
  - ✓ **Guesss.**
    - Adversary A outputs a guess $b' \in \{0,1\}$.
- Advantage
  - ✓ $\varepsilon = |Pr[b' = b] - 1/2|$.

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

ECP-ABE Performance analysis

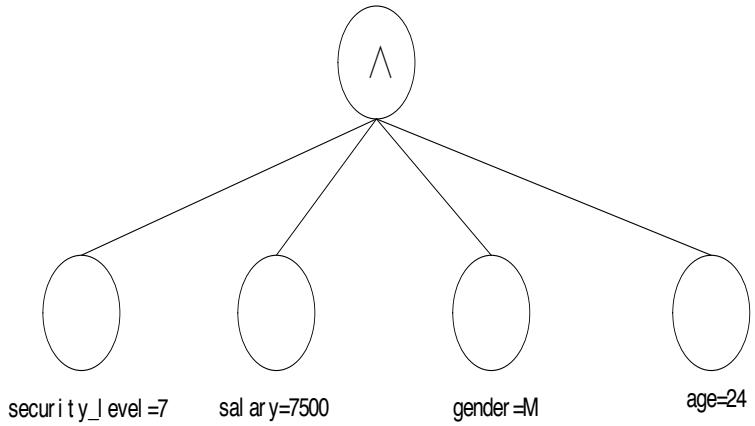
- Security analysis
  - ✓ Message m_b is encrypted under standard tree T.
    - Transformation between T^* and T is public
  - ✓ In **Phase1**, changes of access tree will not introduce any new security problem
    - i.e. the secret key that A gets could not decrypt the ciphertext C_b.
  - ✓ In **Phase1**, Challenge and **Phase2**, adversary A could design the query and challenge against T^*.
  - ✓ So the attacking ability of A keeps the same.

ECP-ABE Performance analysis

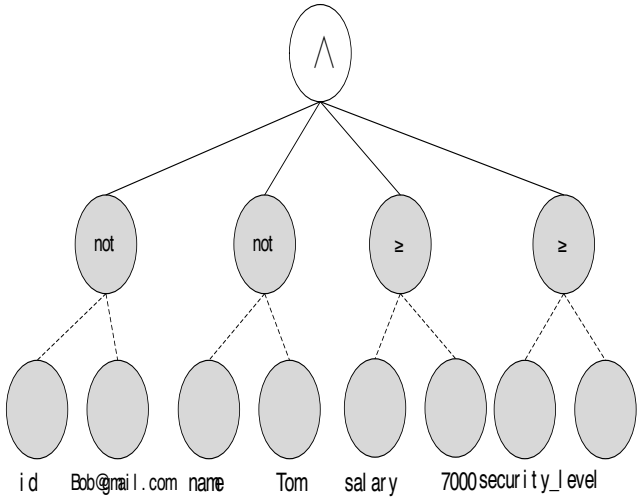
- The added calculation expense:
  - ✓ Encryption phase : the transformation from the extended tree to the standard tree.
  - ✓ Key generation phase: the verification and transformation of the extended attributes.
- Experiment
  - ✓ Two groups of policy file, each group has 10 test policy files
    - One group only contains policies with the standard attributes
    - the other only contains policies with the extended attributes
    - the number of attribute node varies from 1 to 10
  - ✓ Different tiers of structure.
  - ✓ Run three times for each test policy file and get the average cost as the result.

ECP-ABE Performance analysis

- Test case (4 nodes for instance)



security_ level=7 ∧salary=7500∧gender=M∧age=24

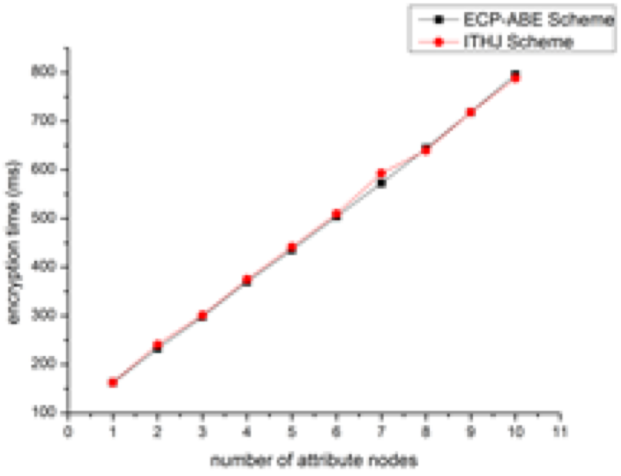


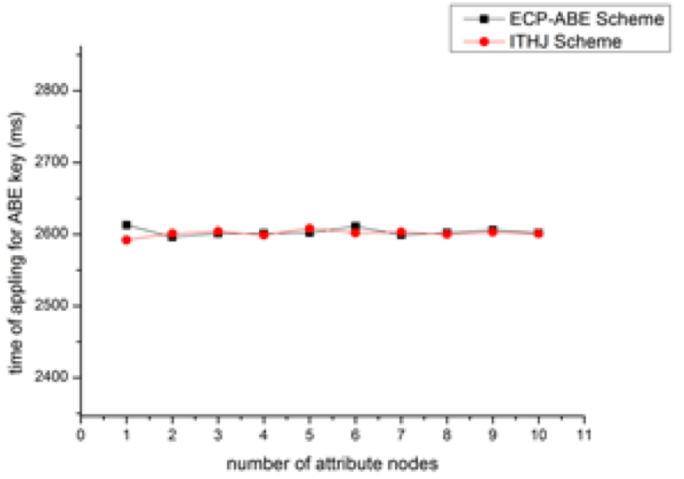
(id not Bob@gmail.com)
∧(name not Tom)
∧(salary≥7000)
∧(security_level≥4)

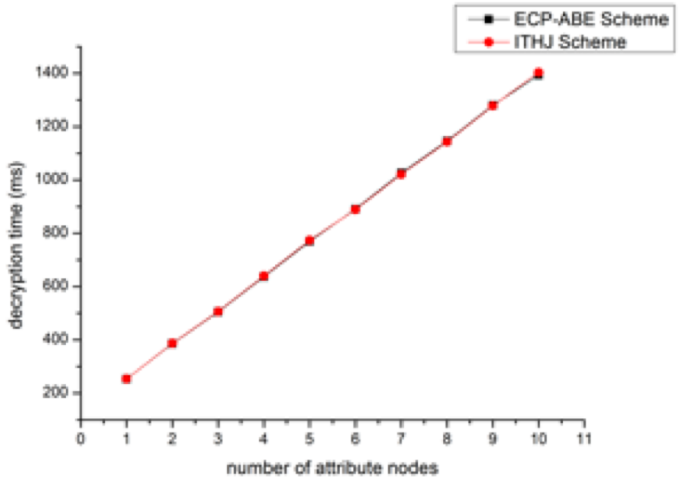
State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

ECP-ABE Performance analysis

- Result







State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

Conclusion

- Propose an ECP-ABE scheme
  - ✓ Introduces the extended leaf nodes into the access policy tree
    - Keep the security and efficiency properties of the CP-ABE scheme
    - Improves the access capability of the baseline scheme
- Future work
  - ✓ Probe other more efficient way to enhance the access control capability of CP-ABE schemes
  - ✓ Design new encryption/decryption algorithms.

State Key Laboratory of Software Development Environment
langbo@buaa.edu.cn

北京航空航天大學
Beihang University

Thank you!
Q & A

langbo@buaa.edu.cn